Методы оценки обобщающей способности

Аннотация

В комбинаторном подходе к проблеме переобучения основной задачей является получение вычислительно эффективных формул для вероятности переобучения и вероятности получить каждый из имеющихся алгоритмов в результате обучения. Предлагается подход, который позволяет проще выводить такие формулы в тех случаях, когда множество алгоритмов наделено некоторой группой симметрий. Приводятся примеры подобных ситуаций. Даётся определение рандомизированного метода обучения, для которого доказывается общая оценка вероятности переобучения.

Содержание

1 Введение

При обучении алгоритмов классификации и прогнозирования по конечным выборкам часто возникает проблема переобучения, когда качество алгоритма, построенного по наблюдаемой обучающей выборке, оказывается значительно хуже на скрытой контрольной выборке.

В работах [1, 2, 4] рассматривался метод *минимизации эмпирического риска*. Он заключается в том, что из заданного множества (семейства) алгоритмов выбирается алгоритм, допускающий наименьшее число ошибок на обучающей выборке.

В следующей таблице показан пример, когда минимизация эмпирического риска приводит к переобучению. Столбцы таблицы соответствуют алгоритмам, строки — объектам генеральной выборки, единица в $[i, d]$-й ячейке таблицы означает, что алгоритм a_d допускает ошибку на объекте x_i. Первые три объекта составляют обучающую выборку, оставшиеся три — контрольную.

	a_1	a_2	...	a_d	...	a_D
x_1	0	1	...	**0**	...	1
x_2	1	1	...	**0**	...	0
x_3	0	0	...	**0**	...	0
x_4	1	1	...	**1**	...	1
x_5	1	0	...	**1**	...	0
x_6	0	0	...	**1**	...	0

В данном примере переобучение могло быть следствием «неудачного» разбиения генеральной выборки на обучение и контроль. Поэтому вводится функционал *вероятности переобучения*, равный доле разбиений выборки, при которых возникает переобучение [4, 3]. Этот функционал инвариантен относительно выбора разбиения и характеризует качество данного метода обучения на данной генеральной выборке.

ToDo Доработать введение.

ToDo Расставить ссылки внутри статьи. Ссылки должны идти «назад», но не «вперед».

1.1 Определения

Пусть задана генеральная выборка $\mathbb{X} = \left(x_i\right)_{i=1}^{L}$, состоящая из L объектов. Произвольный бинарный вектор $a \equiv \left(a(x_i)\right)_{i=1}^{L}$ длины L будем называть *алгоритмом*, и в случае $a(x_i) = 1$ говорить, что алгоритм a допускает ошибку на объекте x_i.

Обозначим через $\mathbb{A} = \{0,1\}^L$ множество всех алгоритмов длины L, тогда под $2^{\mathbb{A}}$ мы будем иметь в виду систему всех возможных множеств алгоритмов. Заметим, что $|\mathbb{A}| = 2^L$, $|2^{\mathbb{A}}| = 2^{2^L}$.

Через $[\mathbb{X}]^\ell$ обозначим множество всех разбиений полной выборки $\mathbb{X}$ на обучающую выборку X^ℓ и контрольную выборку X^k фиксированной длины.

Числом ошибок алгоритма a на выборке $X \subset \mathbb{X}$ назовем величину $n(a, X) = \sum_{x \in X} a(x)$. Под *методом обучения* мы будем иметь в виду некоторое отображение $\mu\colon 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \to \mathbb{A}$. Таким образом метод обучения, имея в своем распоряжении обучающую выборку X^ℓ, выбирает один алгоритм из конечного набора $A \subset \mathbb{A}$. Метод обучения мы будем называть *минимизацией эмпирического риска*, если возвращаемый им алгоритм имеет наименьшее число ошибок на обучении: для всех $X^\ell \in [\mathbb{X}]^\ell$ и $A \in \mathbb{A}$ выполнено $\mu(A, X^\ell) \in \operatorname*{Argmin}_{a \in A} n(a, X)$.

При минимизации эмпирического риска может возникать неоднозначность — несколько алгоритмов могут иметь одинаковое число ошибок на обучающей выборке. В [4] для устранения неоднозначности и получения точных верхних оценок вероятности переобучения использовалась *пессимистичная* минимизация эмпирического риска — предполагалось, что в случае неоднозначности выбирается алгоритм с наибольшим

числом ошибок на генеральной выборке $\mathbb{X}$. Это не устраняет неоднозначность окончательно. Возможны ситуации, когда несколько алгоритмов имеют наименьшее число ошибок на обучающей выборке X^ℓ и одинаковое число ошибок на генеральной выборке $\mathbb{X}$. В таких случаях на множестве алгоритмов вводился линейный порядок, и среди неразличимых алгоритмов выбирался алгоритм с бóльшим номером. Введение приоритетности алгоритмов является искусственным приёмом, не имеющим адекватных аналогов среди известных методов обучения.

1.2 Рандомизированный метод обучения

Формально рандомизированный метод произвольному множеству алгоритмов $A \subset \mathbb{A}$ и произвольной обучающей выборке $X^\ell \in [\mathbb{X}]^\ell$ ставит в соответствие функцию распределения весов на множестве алгоритмов:

$$\mu : 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \to \{f : \mathbb{A} \to [0,1]\}. \tag{1}$$

Предыдущее определение метода обучения $\mu\colon 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \to \mathbb{A}$ получиться, если ограничиться рассмотрением лишь вырожденные функции распределения весов ($f(a)$ равно единице ровно на одном алгоритме, и нулю на всех остальных).

В дальнейшем мы будем называть отображение вида $\mu\colon 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \to \mathbb{A}$ *детерминированным* методом обучения. Вместо определения 1 мы будем пользоваться эквивалентным способом задать то же самое отображение:

$$\mu : 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \times \mathbb{A} \to [0,1].$$

Для большей содержательности дальнейшей теории мы наложим на эту функцию несколько ограничений. В частности, эта функция будет нормирована так, что её можно интерпретировать как вероятность получить данный алгоритм в результате обучения.

Рассмотрим группу S_L — симметрическую группу из L элементов, действующую на множестве объектов выборки перестановками $S_L\colon \mathbb{X} \to \mathbb{X}$.

Для каждого $\pi \in S_L$ определим действие $\pi(X^\ell)$ на произвольную выборку $X^\ell \subset [\mathbb{X}]^\ell$ поэлементным действие отображения $\pi\colon \mathbb{X} \to \mathbb{X}$ на каждый объект выборки X^ℓ: $\pi(X^\ell) = \{\pi(x)\colon x \in X^\ell\}$. Это отображение не меняет числа объектов: $|X^\ell| = |\pi(X^\ell)|$, поэтому можно говорить о действии π на множестве разбиений генеральной выборки на обучение и контроль фиксированной длины $S_L : [\mathbb{X}]^\ell \to [\mathbb{X}]^\ell$.

Определим действие S_L на множестве всех алгоритмов $\mathbb{A}$ перестановкой координат векторов ошибок алгоритмов: $\pi(a) = \big(a(\pi(x_i))\big)_{i=1}^L$. Действие группы S_L на множестве всевозможных алгоритмов $\mathbb{A}$ естественным образом продолжается до действия на системе всех подмножеств — $S_L\colon 2^{\mathbb{A}} \to 2^{\mathbb{A}}$ по правилу $\pi(A) = \{\pi(a)\colon a \in A\}$.

В дальнейшем мы будем пользоваться единым обозначением для описанных выше действий.

Определение 1. Рандомизированным методом обучения *будем называть отображение вида*

$$\mu : 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \times \mathbb{A} \to [0,1]. \tag{2}$$

удовлетворяющее при любых $A \in 2^{\mathbb{A}}$, $X^\ell \in [\mathbb{X}]^\ell$, $a, b \in A$ *и* $\pi \in S_L$ *следующим условиям:*

1. $\sum\limits_{a \in A} \mu(A, X^\ell, a) = 1$;
2. $n(a, X^\ell) = n(b, X^\ell) \;\to\; \mu(A, X^\ell, a) = \mu(A, X^\ell, b)$;
3. $\mu(A, X^\ell, a) = \mu\big(\pi(A), \pi(X^\ell), \pi(a)\big)$.

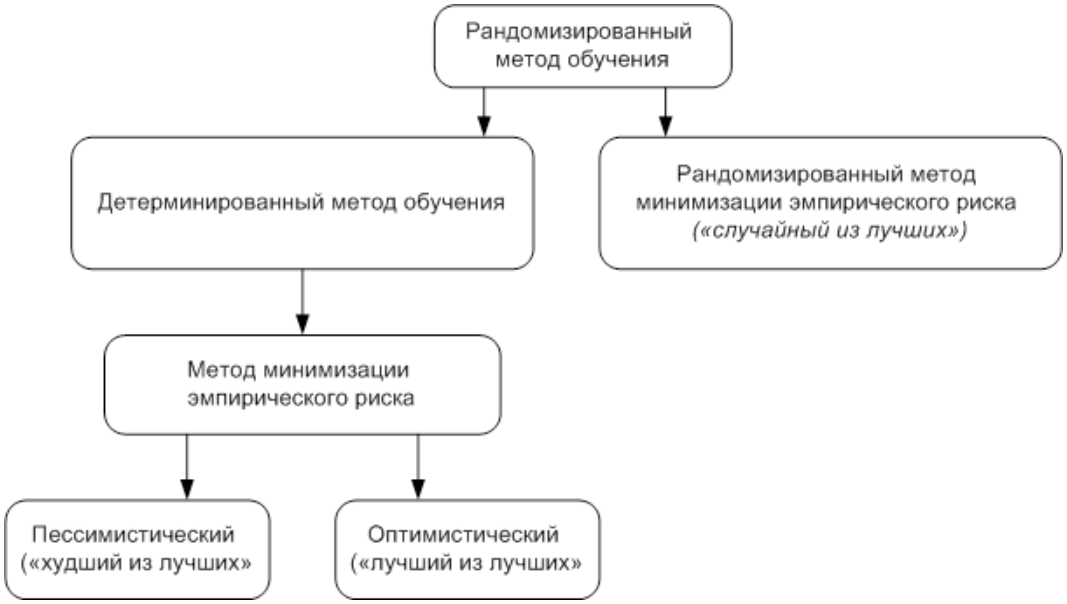

Рис. 1: Методы обучения

Первое условие означает «вероятностную» нормировку весов алгоритмов. Кроме того, оно обеспечивает нулевую «вероятность» алгоритмам, не принадлежащих множеству A. Второе условие означает, что при любом разбиении $(X^\ell, X^k) \in [\mathbb{X}]^\ell$ вероятность получить алгоритм в результате обучения зависит только от количества ошибок алгоритма на обучении. Третье условие означает, что метод обучения не учитывает порядок объектов в выборке.

Нам осталось привести конструктивный пример рандомизированного метода обучения, удовлетворяющего указанному определению. Выделим множество алгоритмов, допускающих минимальное число ошибок на обучающей выборке X:

$$A(X) = \operatorname*{Argmin}_{a \in A} n(a, X).$$

Для всех алгоритмов вне этого множества положим $\mu(A, X, a) = 0$, а алгоритмы из $A(X)$ получат равную вероятность реализоваться:

$$\mu(A, X, a) = \begin{cases} \frac{1}{|A(X)|}, & a \in A(X); \\ 0, & a \notin A(X). \end{cases} \tag{3}$$

Построенное отображение $\mu(A, X, a)$ мы будем называть *рандомизированным методом минимизации эмпирического риска.*

1.3 Вероятность переобучения

Определим функционал вероятности переобучения для рандомизированного метода обучения. Величину $\nu(a, X) = \frac{1}{|X|} n(a, X)$ будем называть *частотой ошибок* алгоритма a на выборке X. *Уклонение частот* на разбиении (X^ℓ, X^k) определим как разность частот ошибок на контроле и на обучении $\delta(a, X^\ell) = \nu(a, X^k) - \nu(a, X^\ell)$.

Зафиксируем параметр $\varepsilon \in (0, 1]$. Будем говорить, что алгоритм a *переобучен* на разбиении (X^ℓ, X^k), если $\delta(a, X^\ell) \geq \varepsilon$.

Вероятностью получить алгоритм $a \in A$ в результате обучения назовем величину

$$P(a, A) = \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} \mu(A, X, a). \tag{4}$$

Для произвольного $\varepsilon \in (0,1]$ определим *вклад* алгоритма $a \in A$ в вероятность переобучения:

$$Q_\mu(a,A) = \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} \mu(A,X,a) \big[\delta(a,X) \geq \varepsilon\big], \tag{5}$$

и саму *вероятность переобучения*:

$$Q_\mu(A) = \sum_{a \in A} Q_\mu(a,A) = \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} \sum_{a \in A} \mu(A,X,a) \big[\delta(a,X) \geq \varepsilon\big]. \tag{6}$$

Для детерминированного метода обучения $\mu\colon 2^{\mathbb{A}} \times [\mathbb{X}]^\ell \to \mathbb{A}$ это определение можно упростить:

$$\begin{aligned} Q_\mu(A) &= \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} \sum_{a \in A} \big[\mu(A,X) = a\big] \big[\delta(a,X) \geq \varepsilon\big] = \\ &= \frac{1}{C_L^\ell} \sum_{X \in [\mathbb{X}]^\ell} \big[\delta(\mu(A,X),X) \geq \varepsilon\big]. \end{aligned}$$

Полученное выражение буквально означает «долю разбиений выборки на обучение и контроль, на которых выбранным методом обучения алгоритм оказался переобученным». В слабой вероятностной аксиоматике принята гипотеза о том, что что все разбиения генеральной выборки на наблюдаемую и скрытую подвыборки равновероятны. Поэтому оператор $\mathbf{P}\left[\varphi(X^\ell)\right] = \frac{1}{C_L^\ell} \sum_{X^\ell \in [\mathbb{X}]^\ell} \left[\varphi(X^\ell)\right]$ следует интерпретировать как вероятность истинности предиката $\varphi(X^\ell)$. Этим объясняется происхождение термина *вероятность переобучения*.

Теорема 1. *Обозначим методы минимизации эмпирического риска: μ — рандомизированный, μ_p — пессимистический, μ_o — оптимистический. Тогда для произвольного множества алгоритмов выполнена следующая цепочка неравенств:*

$$Q_{\mu_o}(A) \leq Q_\mu(A) \leq Q_{\mu_p}(A).$$

Эта теорема позволяет называть методы μ, μ_p и μ_o соответственно выбором случайного, худшего и лучшего алгоритма из лучших на обучении.

□ **Доказательство.** Для краткости обозначений будем опускать аргумент A, считая множество алгоритмов фиксированным. Покажем, что утверждение верно для каждого разбиения выборки:

$$\big[\delta(\mu_o(X),X) \geq \varepsilon\big] \leq \sum_{a \in A} \mu(A,X,a) \big[\delta(a,X) \geq \varepsilon\big] \leq \big[\delta(\mu_p(X),X) \geq \varepsilon\big]$$

Доказательство опирается на очевидные утверждения:

- $F_o = \big[\delta(\mu_o(X),X) \geq \varepsilon\big] \in \{0,1\}$
- $F_p = \big[\delta(\mu_p(X),X) \geq \varepsilon\big] \in \{0,1\}$
- $F_0 = \sum_{a \in A} \mu(A,X,a) \big[\delta(a,X) \geq \varepsilon\big] \in [0,1]$.

Внимательное рассмотрение множества значений этих выражений показывает, что нам достаточно доказать следующие утверждения:

1. $F_o = 1 \longrightarrow F_0 = 1$;
2. $F_p = 0 \longrightarrow F_0 = 0$.

Действительно, пусть метод μ_o выбрал переобученный алгоритм с m_ℓ ошибками на обучении и m_k ошибками на контроле. Пусть рандомизированный метод минимизации эмпирического риска дал не-нулевой вес алгоритмам из множества $A(X)$. Тогда все эти алгоритмы имеют ровно m_ℓ ошибок на обучении, и не менее m_k ошибок на контроле (μ_o, по определению, выбрал лучший алгоритм на контроле среди $A(X)$).

Следовательно, все алгоритмы из $A(X)$ переобучены и импликация $F_o = 1 \longrightarrow F_0 = 1$ выполняется.

В точности аналогичное утверждение показывает справедливость импликации $F_p = 0 \longrightarrow F_0 = 0$. ■

Задача 1. *Привести нетривиальный пример рандомизированного метода обучения, отличный от рандомизированной минимизации эмпирического риска.*

Задача 2. *Привести пример множества алгоритмов, в котором все алгоритмы имеют равную вероятность реализоваться в результате обучения методом минимизации эмпирического риска $P(a, A)$? Можете ли вы доказать свое утверждение? Останется ли справедливым ваше утверждение, если вместо рандомизированной минимизации эмпирического риска рассматривать произвольный рандомизированный метод обучения?*

Задача 3. *В каких пределах может меняться уклонение частот $\delta(a, X^\ell)$?*

1.4 Инвариантность вероятности переобучения к действию группы S_L

Заметим, что нам удалось определить метом минимизации эмпирического риска не опираясь на искусственное введение порядка на множестве алгоритмов. Однако объекты генеральной выборки были пронумерованы. Напомним, что *алгоритм* определялся как бинарный *вектор* $a \equiv \big(a(x_i)\big)_{i=1}^L$, т.е. как *упорядоченная* последовательность нулей и единиц. Однако фактически в определениях рандомизированного метода обучения фигурировало лишь число ошибок. Следовательно, есть шанс надеяться что введенный функционал вероятности переобучения будет инвариантен к изменению нумерации объектов генеральной совокупности.

Лемма 2. *Вероятность получить алгоритм a в результате обучения, а также вероятность переобучения сохраняются при одновременном применении произвольной перестановки $\pi \in S_L$ к множеству A и алгоритму a:*

$$Q(\pi(A)) = Q(A), \tag{7}$$

$$P(\pi(A), \pi(a)) = P(A, a). \tag{8}$$

□ **Доказательство.** Заметим, что применение произвольной перестановки $\pi \in S_L$ к множеству *всех* разбиений выборки на обучение и на контроль $[\mathbb{X}]^\ell$ оставляет это множество на месте: $\pi([\mathbb{X}]^\ell) = [\mathbb{X}]^\ell$. Воспользуемся так же очевидным свойством частоты ошибок алгоритма: $\nu(\pi(a), \pi(X)) = \nu(a, X)$. Тогда

$$\begin{aligned}
Q_\mu(\pi(A)) =& \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \sum_{a \in \pi(A)} \mu(\pi(A), X, a)\, [\delta(a, X) \geq \varepsilon] = \\
=& \frac{1}{C_L^\ell} \sum_{X \in \pi([X]_L^\ell)} \sum_{a \in \pi(A)} \mu(\pi(A), X, a)\, [\delta(a, X) \geq \varepsilon] = \\
=& \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \sum_{a \in A} \mu(\pi(A), \pi(X), \pi(a))\, [\delta(\pi(a), \pi(X)) \geq \varepsilon] = \\
=& \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \sum_{a \in A} \mu(A, X, a)\, [\delta(a, X) \geq \varepsilon] = Q_\mu(A).
\end{aligned}$$

Равенство $P(\pi(A), \pi(a)) = P(A, a)$ получается из выражения $Q_\mu(A) = Q_\mu(\pi(A))$ подстановкой $\varepsilon = -1$. ■

Доказанная только что теорема выглядит очень естественно, поскольку в большинстве задач обучения по прецедентам порядок объектов в выборке действительно не имеет значения. В следующем параграфе мы попробуем построить другие объекты, так же инвариантные к действию группы S_L.

2 Симметрия множества алгоритмов

Попробуем использовать идеи симметрии для получения вычислительно-эффективных формул вероятности переобучения. Что бы понять, какие множества алгоритмов являются «симметричными», а какие — нет, нам необходимо научиться их визуализировать. Для этого мы будем строить *граф смежности* множества алгоритмов — направленный граф $T(A) = (A, E)$, вершины которого соответствуют алгоритмам из A, а ребро $(a_1, a_2) \in E$ соединяет пары алгоритмов, чьи вектора ошибок отличаются только на одном объекте: $\rho(a_1, a_2) = 1$, причем число ошибок алгоритма a_2 на единицу больше, чем у a_1.

2.1 Примеры семейств алгоритмов

Цепочкой алгоритмов будем называть линейно упорядоченное множество алгоритмов, в котором вектор ошибок каждого следующего алгоритма отличается от предыдущего только на одном каком-то объекте.

Монотонной цепочкой называется последовательность алгоритмов, в которой каждый следующий алгоритм допускает ошибки на тех же объектах, что предыдущий, и ещё на каком-то одном объекте.

Связкой из p монотонных цепочек называется множество алгоритмов, полученное объединением p штук монотонных цепочек равной длины («ветвей»), с общим первым алгоритмом, при условии, что множества объектов, на которых ошибаются алгоритмы ветвей, не пересекаются.

Связка из двух ветвей называется *унимодальной цепочкой*. Заметим, что монотонные и унимодальные цепочки можно рассматривать как модели однопараметрических семейств алгоритмов классификации с непрерывной по параметру разделяющей поверхностью [4, 3].

Шаром алгоритмов радиуса r с центром в алгоритме a_0 назовем множество $A_r(a_0) = \{a\colon \rho(a, a_0) \leq r\}$. *Полным m-слоем* назовем множество $a \in \mathbb{A}\colon n(a, X) = m$.

ToDo Добавить определения сеток, картинки и рассуждения. После прочтения данного параграфа у читателя должно сложиться ощущение симметричных и несимметричных ситуаций.

2.2 Группа симметрии множества алгоритмов

Первой на роль объекта, характеризующего свойства симметрии множества алгоритмов, претендует группа автоморфизмов соответствующего графа смежности.

Определение 2. *Группой автоморфизма графа смежности $T(A) = (A, E)$ множества алгоритмов A называют подгруппу $Aut(T(A))$ группы перестановок вершин графа, такую что каждый ее элемент $\pi \in Aut(T(A))$ удовлетворяет двум условиям:*

- *Сохранение ребер графа и их ориентации:*

$$(a_1, a_2) \in E \to \big(\pi(a_1), \pi(a_2)\big) \in E; \tag{9}$$

- *Сохранение числа ошибок алгоритмов:*

$$n(a, \mathbb{X}) = n(\pi(a), \mathbb{X}). \tag{10}$$

Рассматривая унимодальную цепочку легко сделать вывод, что группа автоморфизмов ее графа смежности изоморфна S_2 — группе перестановок из двух элементов. Действительно, из-за условия 10 каждый элемент группы $Aut(T(A))$ должен действовать перестановками пар алгоритмов с равным числом ошибок. После наложения условия 9 получаем, что перестановки алгоритмов в разных ветвях должны производиться *одновременно*, иначе будут нарушены связи между алгоритмами одной ветви.

Задача 4. *Покажите, что для множества алгоритмов со связным графом смежности в приведенном выше определении условие 10 является избыточным.*

В качестве предостережения заметим, что разным семействам алгоритмов может соответствовать один и тот же граф смежности. Пример такой ситуации возможен уже для унимодальной цепочки с ветвями длины 2:

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix} \qquad \begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Для «простой» унимодальной цепочки, изображенной слева, вероятности получить алгоритмы с равным числом ошибок одинаковы. Для унимодальной цепочки «с дефектом» это уже может не выполняться. Этой причины достаточно, что бы продолжить поиски объекта, правильно описывающего свойства симметрии множества алгоритмов. Нам остается воспользоваться стандартным подходом к определению симметрии объекта: найти группу, которая действует на множестве рассматриваемых объектов, и группой симметрии назвать те элементы этой группы, для которых объект является неподвижной точкой.

Аналогия алгоритмов и точек плоскости	
Точка на плоскости	Алгоритм
Плоскость $\mathbb{R}^2$	Множество всех алгоритмов $\mathbb{A}$
Плоская фигура $F \subset \mathbb{R}^2$	Множество алгоритмов $A \subset \mathbb{A}$
Группа поворотов плоскости SO_2	Группа перестановок S_L
$Sym(F) = \{g \in SO_2 \colon g(F) = F\}$	$Sym(A) = \{g \in S_L \colon g(A) = A\}$

Для большей наглядности дальнейших определений проведем следующую аналогию: пусть алгоритмы соответствуют точкам плоскости, множества алгоритмов — плоским фигурам. Зафиксируем некоторую группу преобразований плоскости (например, группу всевозможных движений). Тогда группа симметрии произвольной фигуры определяется как подгруппа, не изменяющая фигуру как множество точек плоскости.

В качестве объемлющей группы преобразований мы возьмем симметрическую группу S_L. Напомним, что выше мы определили действие этой группы на и на генеральную выборку $\mathbb{X}$, и на произвольный алгоритм $a \in \mathbb{A}$, и (поэлементно) на произвольное множество алгоритмов $A \in 2^{\mathbb{A}}$.

Определение 3. Группой симметрий $S(A)$ *множества алгоритмов* $A \in 2^{\mathbb{A}}$ *будем называть его стационарную подгруппу:*

$$S(A) = \{\pi \in S_L \colon \pi(A) = A\}.$$

Каждый элемент группы симметрий $\pi \in S(A)$ переставляет алгоритмы a только *внутри* множества A. Значит, для любого $a \in A$ и любого $\pi \in S(A)$ выполнено $\pi(a) \in A$. Поэтому для группы $S(A)$, в отличии от всей группы S_L, естественным образом определено действие на множестве A.

Орбитой элемента m множества M, на котором действует группа G, называется подмножество $Gm = \{gm \colon g \in G\} \subset M$. Орбиты двух элементов m_1 и m_2 либо не пересекаются, либо совпадают. Это позволяет говорить о разбиении множества M на непересекающиеся орбиты: $M = Gm_1 \sqcup \ldots \sqcup Gm_k$.

Определение 4. *Орбиты действия группы симметрий* $S(A)$ *на множестве алгоритмов* A *будем называть* классами идентичных алгоритмов.

Совокупность всех орбит множества алгоритмов A обозначим через $\Omega(A)$. Представителя орбиты $\omega \in \Omega(A)$ будем обозначать через $a_\omega \in A$. Различных представителей одной и той же орбиты будем называть *идентичными алгоритмами*.

Лемма 3. Идентичные алгоритмы *имеют равное число ошибок на полной выборке.*

Доказательство леммы автоматически следует из очевидного и уже упоминавшегося выше утверждения о сохранении числа ошибок алгоритма при применении к нему произвольной перестановки: $n(a, \mathbb{X}) = n(\pi(a), \mathbb{X})$.

Согласно данному выше определению *алгоритм* $a \equiv \big(a(x_i)\big)_{i=1}^{L}$ является вектором, следовательно, зависит от нумерации объектов выборки. Однако ни группа симметрий $S(A)$, ни разбиение на классы идентичных алгоритмов $\Omega(A)$, уже не зависят от этой нумерации.

Лемма 4. *Для любого множества алгоритмов $A \in 2^{\mathbb{A}}$ и любой перестановки $\pi \in S_L$ группы $S(A)$ и $S(\pi(A))$ сопряжены: $S(\pi(A)) = \pi \circ S(A) \circ \pi^{-1}$.*

Эта лемма эквивалентна известному утверждению из теории групп: стационарные подгруппы точек, лежащих на одной орбите действия, получаются друг из друга сопряжением. Сопряжение устанавливает изоморфизм групп $S(A)$ и $S(\pi(A))$. Остается лишь проверить, что действие изоморфных групп на множествах A и $\pi(A)$ действительно приведет к «одинаковому» разбиению на орбиты.

В следующей таблице приведен пример унимодальной цепочки [5]. Алгоритм a_0 является первым (и наилучшим) в цепочке; a_1, a_2, a_3 составляют левую ветвь; a_4, a_5, a_6 — правую.

	a_0	a_1	a_2	a_3	a_4	a_5	a_6
x_1	0	1	1	1	0	0	0
x_2	0	0	1	1	0	0	0
x_3	0	0	0	1	0	0	0
x_4	0	0	0	0	1	1	1
x_5	0	0	0	0	0	1	1
x_6	0	0	0	0	0	0	1

Перенумерацией объектов выборки ($x_1 \leftrightarrow x_4$, $x_2 \leftrightarrow x_5$, $x_3 \leftrightarrow x_6$) можно поменять левую и правую ветвь местами. Поэтому группой симметрии данного семейства является группа перестановок из двух элементов S_2. Идентичные алгоритмы в унимодальной цепочке — это пары алгоритмов с равным числом ошибок на полной выборке.

Задача 5. *Доказать, что группа симметрии множества алгоритмов A изоморфно вкладывается в группу автоморфизмов соответствующего графа смежности, сохраняющую число ошибок алгоритмов.*

Задача 6. *Доказать, что группа симметрии семейства алгоритмов равна пересечению групп симметрии слоев данного семейства.*

Задача 7. *Найти группу симметрии множества, состоящего из одного алгоритма a с числом ошибок $m = n(a, X)$.*

Задача 8. *Пусть известны группы симметрии множества алгоритмов A_1 и A_2. Найти группу симметрии множества алгоритмов* $A = \begin{pmatrix} A_1 & 0 \\ 0 & A_2 \end{pmatrix}$.

Задача 9. *Найти группу симметрии для следующего множества алгоритмов:*

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Задача 10. *Найти группу симметрии шара алгоритмов* $A_r(a_0) = \{a\colon \rho(a, a_0) \leq r\}$. *Опишите классы идентичных алгоритмов.*

Задача 11. *(?) Согласно теореме Келли любая конечная группа вкладывается в группу перестановок* S_L *(при достаточно большом* L*). Верно ли, что любую конечную группу можно представить как группу симметрии некоторого множества алгоритмов?*

Задача 12. *(?) Сформулируйте условия на множество алгоритмов* A*, при которых группа* $S(A)$ *будет нормальной подгруппой в* S_L*. Предварительно подумайте, можно ли извлечь из этого какую-либо пользу* :)

2.3 Теоремы о равном вкладе идентичных алгоритмов в вероятность переобучения

Ниже мы докажем две общих теоремы о равном вкладе идентичных алгоритмов в вероятность переобучения. В ряде случаев это позволит упростить вывод явных формул для вероятности переобучения.

Теорема 5. *Вероятность получить идентичные алгоритмы в результате обучения одинакова:* $\forall \pi \in S(A)$ *выполнено* $P(\pi(a), A) = P(a, A)$.

□ **Доказательство.**

$$\begin{aligned}
P_\mu(\pi(a), A) &= P_\mu(\pi(a), \pi(A)) = \\
&= \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \mu(\pi(A), X, \pi(a)) = \\
&= \frac{1}{C_L^\ell} \sum_{X \in \pi([X]_L^\ell)} \mu(\pi(A), X, \pi(a)) = \\
&= \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \mu(\pi(A), \pi(X), \pi(a)) = \\
&= \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \mu(A, X, a) = P_\mu(a, A). \ \blacksquare
\end{aligned}$$

Заметим, что нельзя утверждать, что для идентичных алгоритмов a_1 и a_2 при всех $X \in [X]_L^\ell$ выполнено $\mu(A, X, a_1) = \mu(A, X, a_2)$. Речь идет именно о *сумме* вероятности $\mu(A, X, a)$ по всем разбиениям выборки на обучение и контроль.

Следствие 1. *Пусть группа симметрии действует на множестве алгоритмов транзитивно. Тогда все алгоритмы множества имеют равную вероятность реализоваться в результате обучения.*

Напомним, что $\Omega(A)$ — множество классов идентичных алгоритмов, $a_\omega \in A$ — произвольный представитель класса $\omega \in \Omega(A)$.

Теорема 6. *Идентичные алгоритмы дают равный вклад в вероятность переобучения:*

$$Q_\mu(A) = \sum_{\omega \in \Omega(A)} |\,\omega|\, Q_\mu(A, a_\omega). \tag{11}$$

□ **Доказательство.** Распишем вероятность переобучения:

$$\begin{aligned}
Q_\mu(A) =& \frac{1}{C_L^\ell} \sum_{X\in[X]_L^\ell} \sum_{a\in A} \mu(A,X,a)\,[\Delta\nu(a,X)\geq\varepsilon] = \\
=& \frac{1}{C_L^\ell} \sum_{X\in[X]_L^\ell} \sum_{\omega\in\Omega(A)} \sum_{a\in\omega} \mu(A,X,a)\,[\Delta\nu(a,X)\geq\varepsilon] = \\
=& \frac{1}{C_L^\ell} \sum_{\omega\in\Omega(A)} \sum_{a\in\omega} \underbrace{\sum_{X\in[X]_L^\ell} \mu(A,X,a)\,[\Delta\nu(a,X)\geq\varepsilon]}_{N(a)}
\end{aligned}$$

В выделенном в данной формуле выражении $N(a)$ алгоритм a можно заменить на любой идентичный: $N(a) = N(\pi(a))$, где $\pi \in S(A)$. Таким образом вместо суммирования по $a \in \omega$ можно написать $|\omega|$ — количество идентичных алгоритмов. ■

3 Точные оценки вероятности переобучения

В данном параграфе мы будем получать явные комбинаторные формулы для функционала $Q_\mu(\varepsilon, A)$ для различных множеств алгоритмов A. Мы будем опускать аргумент A для сокращения записи, поскольку в каждом конкретном случае будет предварительно описано, о каком множестве алгоритмов идет речь.

Для частоты ошибок алгоритма на полной выборке введем обозначение $m_a = n(a, \mathbb{X})$. Сформулируем лемму, которая является удобным эквивалентным условием условием переобучения $\delta(a, X^\ell) \geq \varepsilon$.

Лемма 7. *Алгоритм a является переобученным на разбиении (X^ℓ, X^k) тогда и только тогда, когда число ошибок на обучении $s = n(a, X^\ell)$ не меньше порога $s_0 = \frac{\ell}{L}(\varepsilon k + m_a)$.*

□ Действительно, согласно определению переобучение алгоритма a записывается неравенством $\frac{m-s}{k} - \frac{s}{\ell} \geq \varepsilon$. Элементарными операциями это неравенство превращается в условие, записанное в лемме. ■

3.1 Вероятность переобучения для одного алгоритма

Рассмотрим вырожденный метод обучения μ_0, который каждой обучающей выборке ставит в соответствие один и тот же алгоритм a_0.

Лемма 8. *Вероятность переобучения метода μ_0, возвращающего фиксированный алгоритм a_0, записывается в виде*

$$Q_{\mu_0}(A) = \frac{H_L^{\ell,m}(s_0)}{C_L^\ell}. \tag{12}$$

Тут $s_0 = \frac{\ell}{L}(\varepsilon k + m_a)$, $H_L^{\ell,m}(s_0) = \sum_{s=s_0}^{\min(m,k)} C_m^s C_{L-m}^{\ell-s}$ — «правый хвост» гипергеометрического распределения.

□ Действительно, вероятность переобучения метода μ_0 определяется числом способов сформировать обучающую выборку длины ℓ так, что бы из m ошибок алгоритма a_0 в X^ℓ оказалось не менее s_0 ошибок. Простое комбинаторное упражнение доказывает, что число способов выбрать ровно s ошибок в обучение записывается в виде числителя гипергеометрического распределения $h_L^{\ell,m}(s) = C_m^s C_{L-m}^{\ell-s}$. Тогда полное число получится если просуммировать $h_L^{\ell,m}(s)$ по всем s от s_0 до максимально-допустимого значения. Заметим, что в наших обозначениях функция $H_L^{\ell,m}(s_0)$ принимает целочисленные значения, отличаясь от классического определения «правого хвоста» гипергеометрического распределения отсутствием знаменателя. Это сделано в целях простоты дальнейших обозначений и лаконичности формул. ■

3.2 Унимодальная цепочка

ToDo Напомнить определение унимодальной цепочки.

Теорема 9. *Для унимодальной цепочки с ветвями длины D вероятность переобучения равна*

$$Q_\mu(\varepsilon) = \frac{1}{C_L^\ell} \sum_{h=0}^{D} \sum_{t_1=h}^{D} \sum_{t_2=0}^{D} \frac{|\omega_h|}{1+S} H_{L'}^{\ell',m}(s(\varepsilon)), \tag{13}$$

где $L' = L-S-F$, $S = t_1+t_2$, $F = [t_1 \neq D]+[t_2 \neq D]$, $\ell' = \ell - F$, $s(\varepsilon) = \left\lfloor \frac{\ell}{L}(m+h-\varepsilon k) \right\rfloor$; $|\omega_h| = 1$ *при* $h = 0$ *и* $|\omega_h| = 2$ *при* $h \geq 1$; $H_{L'}^{\ell',m}(s)$ *— функция гипергеометрического распределения [4].*

□ **Доказательство.**

ToDo Навести порядок в определении гипергеомертического «хвоста». То он суммируется от нуля до s_0, то от s_0 и выше!

Группа симметрий данного множества алгоритмов отождествляет пары алгоритмов с равным числом ошибок. Таким образом вероятность переобучения можно записать в виде

$$Q_\mu(\varepsilon) = \sum_{h=0}^{D} |\omega_h| \sum_{t=h}^{D} \sum_{t'=0}^{D} \sum_{X \in N(t,t')} \mu(A, X, a_h) \left[\Delta\nu(a_h, X) \geq \varepsilon\right].$$

Тут индекс h обозначает номер класса эквивалентных алгоритмов (таким образом, что бы все алгоритмы класса ω_h имели $m+h$ ошибок); $|\omega_h| = 1$ при $h = 0$ и $|\omega_h| = 2$ при $h \geq 1$. Зафиксировав индекс h мы можем считать, что исследуем вероятность переобучения алгоритма a_h, находящегося — для определенности — в левой ветви.

Индексы t и t' параметризуют состав множества лучших на обучении алгоритмов $A_{t,t'}(X)$. Для произвольного разбиения $X \in [\mathbb{X}]_L^\ell$ определим число t как максимальное число, для которого все объекты $x_1, x_2, \ldots, x_t$ находятся в контроле, а x_{t+1} (при его наличии) — в обучении. Индекс t' определяется аналогично, но с помощью объектов правой ветви. Фигурирующее в формуле множество $N(t,t')$ обозначает множество разбиений выборки, каждое из которых приводит к выбору множестве $A_{t,t'}(X)$.

Из определения t и t' следует, что $|A_{t,t'}(X)| = \frac{1}{1+t+t'}$. Единица в знаменателе относится к алгоритму a_0. Индексы t и t' при суммировании пробегают разные множества значений, поскольку при $t < h$ алгоритм a_h не лежит в множестве $A_{t,t'}(X)$, и метод обучения дает ему нулевой вес: $\mu(A, X, a_h) = 0$.

Осталось вычислить число выборок в множестве $N(t,t')$, на которых алгоритм a_h оказывается переобученным. Пусть s_0 — максимальное число ошибок на обучении, при котором переобучение все еще наблюдается. По аналогии с леммой 8 находим $s_0 = \lfloor \frac{\ell}{L}(m+h-\varepsilon k) \rfloor$.

Обозначим $L' = L - t' - t - 2$, $\ell' = \ell - 2$.

Нам необходимо из L' объектов выбрать ℓ' для обучения таким образом, что бы из m свободных ошибок алгоритма a_h в обучении оказалось не более s_0 ошибок. По определению «хвоста» гипергеометрического распределения это можно сделать $\sum_{s=0}^{s_0} C_m^s C_{L'-m}^{\ell'-s}$ способами.

Собирая все результаты, приходим к окончательной формуле:

$$Q_\mu(\varepsilon) = \sum_{h=0}^{D} |\omega_h| \sum_{t=h}^{D} \sum_{t'=0}^{D} \frac{1}{1+t+t'} \frac{\sum_{s=0}^{s_0} C_m^s C_{L'-m}^{\ell'-s}}{C_L^\ell}.$$

В данной формуле мы пренебрегли эффектами, связанными с нижними краями цепочки. В действительности следовало бы учесть, что $\ell' = \ell - [t \neq D] - [t' \neq D]$, и аналогично для L'. ■

Заметим, что данное рассуждение легко распространяется на произвольное количество «хвостов» у цепочек.

3.3 Связка из монотонных цепочек

Напомним, что *связкой из p монотонных цепочек* называется множество алгоритмов, полученное объединением p штук монотонных цепочек равной длины («ветвей»), с общим первым алгоритмом, при условии, что множества объектов, на которых ошибаются алгоритмы ветвей, не пересекаются.

Нетрудно установить, что группа симметрии связки из p монотонных цепочек является симметрической группой S_p, действующей на ветви связки всевозможными перестановками. Таким образом, классы идентичных алгоритмов — это подмножества алгоритмов с одинаковым числом ошибок на полной выборке, называемые *слоями* [4].

В следующей теореме мы получим явную формулу вероятности переобучения для связки из p монотонных цепочек. При этом нам понадобится комбинаторный коэффициент $R^h_{D,p}(S,F)$, который зависит от параметров S и F, от числа монотонных цепочек p и от их длины D, а также от h — минимального значения параметра S. Коэффициент $R^h_{D,p}(S,F)$ равен числу способов представить число S в виде суммы p неотрицательных слагаемых, $S = t_1 + \ldots + t_p$, каждое из которых не превосходит D. При этом ровно F слагаемых не должно равняться D, а на первое слагаемое накладывается дополнительное ограничение $t_1 \geq h$.

Теорема 10. *Рассмотрим связку из p монотонных цепочек, в которой лучший алгоритм допускает m ошибок на полной выборке, длина каждой ветви без учета лучшего алгоритма — D. Тогда при обучении рандомизированным методом вероятность переобучения может быть записана в виде:*

$$Q_\mu(\varepsilon) = \sum_{h=0}^{D} \sum_{S=h}^{pD} \sum_{F=0}^{p} \frac{|\omega_h| R^h_{D,p}(S,F)}{1+S} \frac{C^{\ell'}_{L'}}{C^{\ell}_{L}} H^{\ell',m}_{L'}(s(\varepsilon)), \tag{14}$$

где $L' = L-S-F$, $\ell' = \ell - F$, $s(\varepsilon) = \left\lfloor \frac{\ell}{L}(m+h-\varepsilon k) \right\rfloor$; $|\omega_h| = 1$ при $h = 0$ и $|\omega_h| = p$ при $h \geq 1$; $H^{\ell',m}_{L'}(s)$ — функция гипергеометрического распределения [4].

□ **Доказательство.** Естественным образом обобщая рассуждения, приведенные для унимодальной цепочки, получаем формулу

$$Q_\mu(\varepsilon) = \sum_{h=0}^{D} |\omega_h| \sum_{t_1=h}^{D} \sum_{t_2=0}^{D} \ldots \sum_{t_p=0}^{D} \frac{1}{1+t_1+t_2+\ldots+t_p} \frac{\sum_{s=0}^{s_0(\varepsilon)} C^s_m C^{\ell'-s}_{L'-m}}{C^{\ell}_{L}},$$

где $L' = L - \sum_{i=1}^{p} t_i - \sum_{i=1}^{p} [t_i \neq D]$, $\ell' = \ell - \sum_{i=1}^{p} [t_i \neq D]$, $s_0(\varepsilon) = \lfloor \frac{\ell}{L}(m+h-\varepsilon k) \rfloor$.

Запишем формулу с помощью гипергеометрического распределения:

$$Q_\mu(\varepsilon) = \frac{1}{C^{\ell}_{L}} \sum_{h=0}^{D} |\omega_h| \sum_{t_1=h}^{D} \sum_{t_2=0}^{D} \ldots \sum_{t_p=0}^{D} \frac{1}{1+t_1+t_2+\ldots+t_p} H^{\ell',m}_{L'}(s_0(\varepsilon)),$$

Упростим запись, введя дополнительные обозначения $S = \sum_{i=1}^{p} t_i$, $F = \sum_{i=1}^{p} [t_i \neq D]$. Параметр S, при таком определении, будет задавать мощность множества лучших на обучении алгоритмов $A(X)$.

$$Q_\mu(\varepsilon) = \frac{1}{C^{\ell}_{L}} \sum_{h=0}^{D} |\omega_h| \sum_{t_1=h}^{D} \sum_{t_2=0}^{D} \ldots \sum_{t_p=0}^{D} \frac{1}{1+S} H^{\ell',m}_{L'}(s_0(\varepsilon)),$$

где $L' = L - S - F$, $\ell' = \ell - F$, $s_0 = \lfloor \frac{\ell}{L}(m+h-\varepsilon k) \rfloor$.

Теперь можно перейти к суммированию по множеству возможных значений параметров S и F.

$$Q_\mu(\varepsilon) = \frac{1}{C^{\ell}_{L}} \sum_{h=0}^{D} |\omega_h| \sum_{S=h}^{pD} \sum_{F=0}^{p} \frac{R^h_D(S,F,p)}{1+S} H^{\ell',m}_{L'}(s_0(\varepsilon)),$$

Здесь $R^h_D(S,F,p)$ — комбинаторный коэффициент, определенный в формулировке теоремы. ■

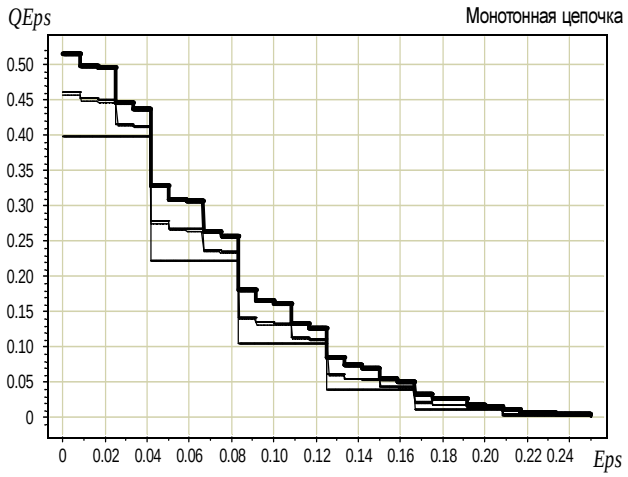

Рис. 2: Зависимость $Q_\mu(\varepsilon)$ от ε для монотонной цепочки при $L = 100$, $\ell = 60$, $D = 40$, $m = 20$.

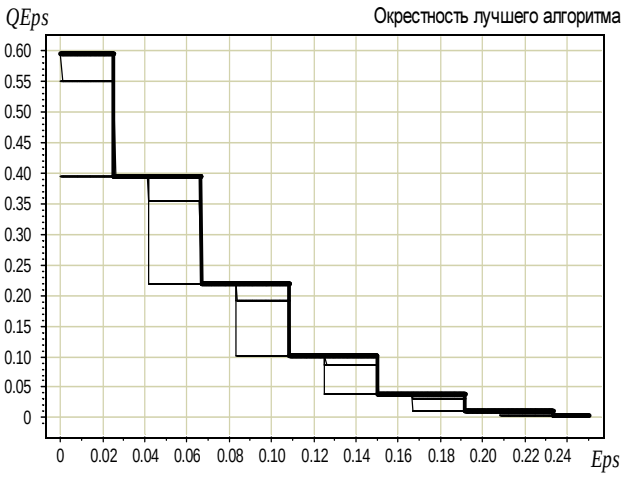

Рис. 3: Зависимость $Q_\mu(\varepsilon)$ от ε для единичной окрестности при $L = 100$, $\ell = 60$, $p = 10$, $m = 20$.

3.3.1 Оценка для монотонной цепочки и единичной окрестности

Связка из p монотонных цепочек является обобщением трёх частных случаев, рассмотренных в [3]: монотонной цепочки ($p = 1$), унимодальной цепочки ($p = 2$) и единичной окрестности лучшего алгоритма ($D = 1$). Вычисляя конкретные выражения комбинаторного коэффициента $R^h_{D,p}(S, F)$ для этих случаев, получим два оставшихся следствия.

Следствие 2. *Для монотонной цепочки длины $D + 1$ вероятность переобучения равна*

$$Q_\mu(\varepsilon) = \sum_{h=0}^{D} \sum_{S=h}^{D} \frac{1}{1+S} \frac{C^{\ell'}_{L'}}{C^{\ell}_{L}} H^{\ell',m}_{L'}(s(\varepsilon)), \tag{15}$$

где $L' = L - S - [S{\neq}D]$, $\ell' = \ell - [S{\neq}D]$.

Следствие 3. *Для единичной окрестности из $p + 1$ алгоритма вероятность переобучения равна*

$$Q_\mu(\varepsilon) = \sum_{h=0}^{1} \sum_{S=h}^{p} \frac{|\omega_h| C^{S-h}_{p-h}}{1+S} \frac{C^{\ell'}_{L'}}{C^{\ell}_{L}} H^{\ell',m}_{L'}(s(\varepsilon)), \tag{16}$$

где $L' = L{-}p$, $\ell' = \ell{+}S{-}p$.

3.3.2 Численный эксперимент

На рис. 2 и 3 представлены результаты численных экспериментов. Из четырех кривых на каждом графике верхняя (жирная) соответствует пессимистической минимизации эмпирического риска [3, 4], нижняя — оптимистической. Две почти сливающиеся кривые между ними соответствуют рандомизированной минимизации эмпирического риска. Одна из них вычислена по доказанным формулам, вторая построена методом Монте-Карло по 10^5 случайных разбиений, при равновероятном выборе лучшего алгоритма в случаях неопределенности. Совпадение этих двух кривых подтверждает справедливость развитой выше теории.

3.4 Полный слой алгоритмов

Рассмотрим множество, состоящее из всех алгоритмов с фиксированным числом ошибок: $a \in \mathbb{A}\colon n(a,X) = m$. Такое множество мы договорились называть *полным m-слоем* алгоритмов.

Теорема 11. *При обучении рандомизированным методом минимизации эмпирического риска вероятность переобучения для полного m-слоя алгоритмов определяется выражением*

$$Q_\mu(\varepsilon) = \begin{cases} 1, & \text{при } \varepsilon k \le m \le L - \varepsilon\ell; \\ 0, & \text{в противном случае.} \end{cases} \tag{17}$$

□ **Доказательство.**

В рассматриваемом случае группой симметрии $S(A)$ будет вся симметрическая группа S_L. Следовательно, действие группы симметрии на множестве алгоритмов транзитивно, и мы имеем дело с одним классом из C_L^m идентичных алгоритмов. Согласно теореме о равном вкладе получаем

$$Q_\mu(\varepsilon) = \frac{C_L^m}{C_L^\ell} \sum_{X \in [X^\ell]} \mu(A, X, a_0) \big[\delta(a_0, X) \ge \varepsilon\big].$$

где a_0 — произвольный алгоритм рассматриваемого семейства.

Благодаря свойствам рассматриваемого семейства алгоритм a_0 будет выбран только если он имеет минимально-возможное число ошибок на обучении. Рассмотрим два случая.

Случай 1, $m \le k$. Тогда все ошибки помещаются в контроль, и переобучение будет наступать в том случае если $m \ge \varepsilon k$. Этим зафиксированы m объектов контроля, и нам необходимо выбрать дополнительно C_{L-m}^{k-m} объектов, на которых алгоритм a_0 не ошибается. Теперь, согласно определению рандомизированного метода минимизации эмпирического риска 3, нам нужно определить состав множества лучших на обучении алгоритмов $A(X)$. В каждом случае в множестве $A(X)$ будет оказываться C_k^m алгоритмов. Все эти алгоритмы не должны иметь ошибок на обучении, поэтому данное число определяется количеством способов расставить m ошибок алгоритма на k позиций контрольной выборки. Итого получаем

$$Q_\mu(\varepsilon) = \frac{C_L^m}{C_L^\ell} \frac{C_{L-m}^{k-m}}{C_k^m} \left[m \ge \varepsilon k\right], \text{ при } m \le k.$$

Случай 2, $m > k$. Тогда в обучении останется $m-k$ ошибок, а условие переобучения имеет вид $1 - \frac{m-k}{\ell} \ge \varepsilon$. Т.е. $m \le L - \varepsilon\ell$.

Число разбиений выборки, при которых алгоритм a_0 получает в результате обучения ненулевой вес, равно C_m^k (число способов выбрать k ошибок алгоритма a_0, расположенных в контрольной выборке). Вместе с алгоритмом a_0 в множестве $A(X)$ будут присутствовать C_ℓ^{m-k} алгоритмов (число способов выбрать $m-k$ ошибок, попавших в обучающую выборку). Итого получаем

$$Q_\mu(\varepsilon) = \frac{C_L^m}{C_L^\ell} \frac{C_m^k}{C_\ell^{m-k}} \left[m \le L - \varepsilon\ell\right], \text{ при } m > k.$$

Расписывая все комбинаторные коэффициенты как $C_L^k = \frac{L!}{k!(L-k)!}$ убеждаемся, что комбинаторные множители в обеих формулах равны единице! Соединяя вместе два условия $\varepsilon k \le m \le k$ и $k < m \le L = \varepsilon\ell$ получаем утверждение теоремы. ■.

4 Универсальные верхние оценки вероятности переобучения

4.1 Принцип равномерной сходимости

Что бы получить верхние оценки вероятности переобучения для произвольного метода обучения μ, в теории Вапника-Червоненкиса [?, ?] и в ряде последующих работ вводится *принцип равномерной сходимости*. Функционал $Q_\mu(\varepsilon, A)$ заменяется его верхней оценкой — вероятностью *наибольшего* уклонения частот в двух подвыборках:

$$Q_\mu(\varepsilon, A) = \mathbf{P}\left[\delta_\mu(A, X^\ell) \geq \varepsilon\right] \leq \mathbf{P}\left[\max_{a \in \mu(A)} \delta(a, X^\ell) \geq \varepsilon\right]. \tag{18}$$

Тут под $\mu(A)$ нужно понимать множество всех алгоритмов, которые могут быть выбраны методом обучения: $\mu(A) = \{\mu(A, X^\ell) \,|\, X^\ell \in [\mathbb{X}]^\ell\}$.

Величину, стоящую в правой части выражения, принято оценивать с помощью неравенства Буля: вероятность объединения событий оценивается мажорируется суммой вероятностей отдельных событий. Такие оценки являются сильно завышенными, и потому редко применимыми на практике.

В качестве интересного исключения можно назвать следующую теорему:

Теорема 12. ***ToDo*** *Добавить ссылку на статью Воронцова.*

Если все алгоритмы из рассматриваемого множества имеют равное число ошибок на полной выборке, а в качестве метода обучения используется минимизация эмпирического риска, оценка 18 обращается в точное равенство.

Задача 13. *Получить теорему 11 для детерминированного метода минимизации эмпирического риска. Подсказка: воспользоваться принципом равномерной сходимости и предыдущей теоремой.*

4.2 Профиль расслоения-связности $D_A(m, q)$

Для каждого алгоритма $a \in A$ обозначим через E_a множество объектов генеральной выборки $\mathbb{X}$, на которых он допускает ошибку: $E_a = \{x_i \in \mathbb{X} : a(x_i) = 1\}$. Очевидно, $n(a, \mathbb{X}) = |E_a|$.

Подмножество алгоритмов $A_m = \{a \in A \colon n(a, \mathbb{X}) = m\}$ называется *m-м слоем* множества A. Разбиение множества A на слои $A = \sqcup_{m=0}^{L} A_m$ называется *расслоением* множества алгоритмов A. Профилем расслоения $D_A(m)$ называется функция, значения которой определяются количеством алгоритмов с фиксированным числом ошибок: $D_A(m) = |A_m|$.

Теорема 13. *[?] Для произвольного метода обучения μ, генеральной выборки $\mathbb{X}$ и числа $\varepsilon \in (0, 1]$ справедлива оценка*

$$Q_\mu(\varepsilon, A) \leq \sum_{m=0}^{L} D_m \frac{H_L^{\ell,m}(s_0)}{C_L^\ell}. \tag{19}$$

Связностью $q(a)$ алгоритма $a \in A$ будем называть число алгоритмов в следующем слое, допускающих ошибки на тех же объектах, что и a:

$$q(a) = \#\left\{a' \in A \colon |E_{a'} \backslash E_a| = 1, |E_{a'}| - |E_a| = 1\right\}. \tag{20}$$

Связность $q(a)$ алгоритма a — это число рёбер графа смежности, исходящих из вершины a.

Профилем расслоения-связности $D_A(m, q)$ для семейства алгоритмов A назовем функцию, значения которой соответствуют числу алгоритмов в m-слое, имеющих связность q.

Теорема 14. *[?] Пусть в A есть корректный на $\mathbb{X}$ алгоритм. Тогда для пессимистической минимизации эмпирического риска справедлива верхняя оценка вероятности переобучения*

$$Q_\mu(\varepsilon, A) \leq \sum_{m=\lceil \varepsilon k \rceil}^{L} \sum_{q=0}^{L} D_A(m,q) \frac{C_{L-m-q}^{\ell-q}}{C_L^\ell}. \tag{21}$$

В работе [?] данная теорема доказывается с помощью сложной рекуррентной процедуры. Ниже мы предложим альтернативное доказательство этой теоремы.

4.3 Теорема о производящих и разрушающих объектах

ToDo. Причесать обозначения в этом параграфе.

Зафиксируем множество алгоритмов A и метод обучения μ. Допустим что для каждого алгоритма $a \in A$ удалось в явном виде указать множество разбиений $U_a = \{(X_i^\ell, X_i^k)\}$, при которых алгоритм a результатом обучения: $a = \mu(X_i^\ell)$:

Гипотеза 1. *Пусть для каждого алгоритма $a \in A$ можно указать пару непересекающихся множеств $X_a \subset \mathbb{X}$ и $X'_a \subset \mathbb{X}$, таких что*

$$[\mu(X^\ell) = a] = [X_a \subset X^\ell][X'_a \subset X^k] \tag{22}$$

Гипотеза 22 означает, что для каждого алгоритма a существует множество *эталонных* объектов X_a, которые обязаны присутствовать в обучающей выборке X^ℓ, и множество *шумовых* объектов X'_a, которых не должно быть в обучающей выборке, что бы метод μ выбрал алгоритма a. Все остальные объекты $\mathbb{X}\backslash(X_a \cup X'_a)$ будем называть *нейтральными* для алгоритма a. Их присутствие в обучающей выборке не влияет на результат обучения. В работах [?, 5] приведены примеры нетривиальных семейств алгоритмов, для которых гипотеза 22 выполняется.

Введем следующие обозначения:

$L_a = L - |X_a| - |X'_a|$ — число объектов, нейтральных для a.

$\ell_a = \ell - |X_a|$ — число объектов, нейтральных для a, в обучающей выборке X^ℓ.

$k_a = k - |X'_a|$ — число объектов, нейтральных для a, в контрольной выборке X^k.

Лемма 15. *[?] Если гипотеза 22 справедлива, то вероятность получить в результате обучения алгоритм a равна*

$$P_a = \boldsymbol{P}[\mu(X^\ell) = a] = \frac{C_{L_a}^{\ell_a}}{C_L^\ell}. \tag{23}$$

Теорема 16. *[?] Если гипотеза 22 справедлива, то вероятность переобучения вычисляется по формуле*

$$Q_\mu(\varepsilon, A) = \sum_{a \in A} \frac{H_{L_a}^{\ell_a, m_a}(s_a)}{C_L^\ell}. \tag{24}$$

Теорема 16 позволяет выписывать точные оценки вероятности переобучения, однако оставляет открытым вопрос о множестве выборе множества эталонных и шумовых объектов. При этом гипотеза 22 требует, что бы условие было необходимым и достаточным. Однако уже для унимодальных цепочек и сеток предъявить подобный критерий получения алгоритма не просто. Заметим, что ослабив требования теоремы, мы получим простую и универсальную верхнюю оценку для вероятности переобучения.

Следствие 4. *Пусть $[X_a \subset X^\ell]$ и $[X'_a \subset X^k]$ будут необходимыми, но уже не обязательно достаточным условием получения алгоритма в результате обучения. Тогда явная формула предыдущей теоремы будет давать верхнюю оценку вероятности переобучения.*

□ Доказательство. **ToDo.** ■

Предъявим конструктивный способ построения множеств X_a и X'_a для произвольного множества алгоритмов и *пессимистического* метода минимизации эмпирического риска μ_p. **ToDo.** Сделать иллюстрацию (на примере унимодальной цепочки).

Множество $\bar{A}_a \subset \mathbb{A}$ — это множество алгоритмов, чье множество ошибок содержится среди множества ошибок алгоритма a. Это — все строго более хорошие алгоритмы. Определим множество всех объектов, на которых алгоритм a ошибается, а хотя бы один $a' \in \bar{A}_a$ — нет:

$$X'_a = \bigcup_{a' \in \bar{A}_a} a \backslash a'$$

Для выбора алгоритма a методом пессимистической минимизации эмпирического риска необходимо, что бы выполнялось $X'_a \subset X^k$.

Аналогично построим множество строго более плохих алгоритмов, делающих ровно на **одну** ошибку больше: $A_a \in \mathbb{A}$ и множество всех объектов, на которых a не ошибается, а хотя бы один из $a' \in A_a$ — ошибается:

$$X_a = \bigcup_{a' \in A_a} a' \backslash a$$

Для $\mu_p(X) = a$ необходимо что бы выполнялось $X_a \subset X$.

Заметим, что между построенными множествами нет естественной симметрии — множество X_a зависит только от строго на единицу худших алгоритмов, в то время как X'_a зависит только от всех строго лучших алгоритмов.

Таким образом, у нас есть универсальный способ построения множеств разрушающий и производящих объектов для произвольных семейств, и следствие теоремы, которое утверждает, что полученная оценка будет верхней. Легко доказать, что данная оценка является точной для монотонной цепочки. В работе [5] доказывается, что приведенная выше оценка будет точна и для многомерной монотонной сетки алгоритмов произвольной размерности.

С помощью полученной оценки легко доказать теорему 21. Действительно, пусть в множестве алгоритмов есть корректный. Тогда каждый алгоритм с m-ошибками имеет минимум m объектов в множестве X'_a. Согласно определению профиля связности q — это и есть число объектов в построенном нами множестве X_a. Тогда $L'_a = L - m - q$, $\ell'_a = \ell - q$ и наша оценка запишется так

$$\begin{aligned} Q_\mu(\varepsilon, A) \leq \sum_{a \in A} \frac{H^{\ell-q,m}_{L-m-q}(s_a)}{C^\ell_L} = \sum_{m=\lceil \varepsilon k \rceil}^{L} \sum_{q=0}^{L} D_A(m,\, q) \frac{H^{\ell-q,m}_{L-m-q}(s_a)}{C^\ell_L} \leq \\ \leq \sum_{m=\lceil \varepsilon k \rceil}^{L} \sum_{q=0}^{L} D_A(m,\, q) \frac{C^{\ell-q}_{L-m-q}}{C^\ell_L}. \end{aligned} \tag{25}$$

В последнем неравенстве мы воспользовались очевидной оценкой для гипергеометрического распределения $H^{\ell,m}_L(s_0) \leq C^\ell_L$.

4.4 Оценки, основанные на неравенстве Коши-Буняковского

Для того, что бы выписать точную оценку вероятности переобучения для конкретного семейства необходимо для каждого алгоритма указать множество условий, накладываемых на разбиение на обучение и контроль, при которых данный алгоритм имеет шанс реализоваться. При этих ограничениях необходимо посчитать число случаев, когда выполнено $[\Delta\nu(a, X) \geq \varepsilon]$. Это число уже может не выражаться в виде простого комбинаторного выражения. В идеале хотелось бы иметь оценку, в которую для каждого алгоритма входит вероятность получить его в результате обучения и вероятность переобучения данного алгоритма на всей выборке, зависящая только от числа ошибок алгоритма.

Подобную оценку можно получить с помощью неравенства Коши-Буняковского-Шварца.

$$\begin{aligned}
Q_\mu(\varepsilon, A) &= \frac{1}{C_L^\ell} \sum_{X \in [X]_L^\ell} \sum_{a \in A} \mu(A, X, a) \left[\Delta\nu(a, X) \geq \varepsilon\right] = \\
&= \frac{1}{C_L^\ell} \sum_{a \in A} \sum_{X \in [X]_L^\ell} \mu(A, X, a) \left[\Delta\nu(a, X) \geq \varepsilon\right] \leq \\
&\leq \sum_{a \in A} \sqrt{\frac{\sum_{X \in [X]_L^\ell} \mu^2(A, X, a)}{C_L^\ell}} \sqrt{\frac{\sum_{X \in [X]_L^\ell} \left[\Delta\nu(a, X) \geq \varepsilon\right]^2}{C_L^\ell}} = \\
&= \sum_{a \in A} \sqrt{\frac{\sum_{X \in [X]_L^\ell} \mu^2(A, X, a)}{C_L^\ell}} \sqrt{H_L^{\ell, m_a}(s)} = \\
&= \sum_{a \in A} \sqrt{\langle \mu^2(A, X, a) \rangle} \sqrt{H_L^{\ell, m_a}(s)}
\end{aligned} \tag{26}$$

В итоговую оценку входит среднеквадратичная вероятность получения алгоритма в результате обучения и корень из гипергеометрического распределение на полной выборке. Данный результат легко совместить с теоремой о равном вкладе эквивалентных алгоритмов в вероятность переобучения:

$$Q_\mu(\varepsilon, A) \leq \sum_{\omega \in \Omega(A)} |\omega| \sqrt{\langle \mu^2(A, X, a_\omega) \rangle} \sqrt{H_L^{\ell, m_{a_\omega}}(s)} \tag{27}$$

Можно сформулировать этот результат для метода обучения, который возвращает один алгоритм: $\mu : 2^{\mathbb{A}} \times [X]_L^\ell \to \mathbb{A}$:

$$\begin{aligned}
Q_\mu(\varepsilon, A) &= \frac{1}{C_L^\ell} \sum_{a \in A} \sum_{X \in [X]_L^\ell} \left[\mu(A, X) = a\right] \left[\Delta\nu(a, X) \geq \varepsilon\right] = \\
&\leq \sum_{a \in A} \sqrt{\frac{\sum_{X \in [X]_L^\ell} \left[\mu(A, X) = a\right]^2}{C_L^\ell}} \sqrt{\frac{\sum_{X \in [X]_L^\ell} \left[\Delta\nu(a, X) \geq \varepsilon\right]^2}{C_L^\ell}} = \\
&= \sum_{a \in A} \sqrt{P_a} \sqrt{H_L^{\ell, m_a}(s)} = \sum_{a \in A} \sqrt{P_a H_L^{\ell, m_a}(s)}.
\end{aligned} \tag{28}$$

Обозначим бинарный вектор $[\mu(A, X) = a]$ длиной C_L^ℓ через $\overline{s}_a$, а вектор $[\Delta\nu(a, X) \geq \varepsilon]$ — через $\overline{o}_a$. Тут s — от «selected», o — от «overfited». Интуитивно очевидно, что именно выбранные методом минимизации эмпирического риска алгоритмы имеют больше всего шансов оказаться переобученными. Таким образом есть основания полагать, что вектора $\overline{s}_a$ и $\overline{o}_a$ «кореллируют». А именно коллинеарность векторов является условием, при котором оценка неравенства Коши-Буняковского обращается в точное равенство.

Возможно, указанное выше обстоятельство позволит использовать приведенную оценку на практике в качестве внешнего критерия для отбора методов обучения.

5 Заключение

Свойство симметрии семейств алгоритмов позволяет упрощать получение вычислительно эффективных формул вероятности переобучения. В частности, удалось вывести оценки для монотонной и унимодальной цепочек, а также для единичной окрестности лучшего алгоритма как следствия одной общей теоремы, в то время как ранее аналогичные оценки доказывались независимо и при неестественном предположении об априорной упорядоченности алгоритмов в семействе.

ToDo. Доработать выводы.

Список литературы

[1] *Вапник В. Н., Червоненкис А. Я.* Теория распознавания образов. — М.: Наука, 1974.

[2] *Vapnik V.* Statistical Learning Theory. — New York: Wiley, 1998.

[3] *Воронцов К. В.* Точные оценки вероятности переобучения // Доклады РАН, 2009 (в печати).

[4] *Воронцов К. В.* Комбинаторный подход к проблеме переобучения // Всеросс. конф. ММРО-14 — М. МАКС Пресс, 2009.

[5] *Ботов П. В.* Точные оценки вероятности переобучения для монотонных и унимодальных семейств алгоритмов // Всеросс. конф. ММРО-14 — М. МАКС Пресс, 2009.

[6] *Винберг Э. Б.* Курс алгебры // М.: Факториал Пресс, 2001. — 544 с.